Tracking Appliance Usage Information Using Harmonic Signature Sensing

Deokwoo Jung*† Hoang Hai Nguyen * David K. Y. Yau*‡
{deokwoo.jung,hoanghai.david.yau}@adsc.com.sg
†Corporate R&D Center, SK Telecom, Seoul, Republic of Korea
deokwoo.jung@sk.com
‡Singapore University of Technology and Design
david_yau@sutd.com.sg

Abstract—Real-time usage of individual electrical appliances is a key enabler of important advanced services for smart grids. With wide deployments of smart meters, there is a growing interest in using Non-Intrusive Load Monitoring (NILM) to acquire this information from the meter measurements. However, electrical signatures extracted from utility-side smart meters are often unreliable for NILM due to their large sampling intervals. This paper presents a new approach of using high-frequency current waveforms sampled periodically at a main branch to track reliably the on/off states of appliances in real-time. We develop an incremental training algorithm and a robust detection algorithm for the harmonic signatures, based on semi-supervised learning and a hidden Markov model, respectively. We evaluate the performance of the training and detection algorithms using simulations and a proof-of-concept testbed with five appliances. The simulation results show that our state detection algorithm is highly robust against noisy harmonic signatures – up to 16 times more robust than a baseline algorithm without the hidden Markov model. The experimental results show that the proposed algorithms can successfully learn most harmonic signatures using only 10% of label information. They can detect the on/off states with less than 4% errors.

I. INTRODUCTION

Increasing deployment of smart meters in residential, commercial, and industrial buildings brings about useful intelligence in emerging smart electrical grids. In particular, Non-Intrusive Load Monitoring (NILM) [4] is gaining attention. NILM is a load disaggregation technique that extracts unique electrical signatures of specific end loads from aggregate power measurements at a main branch. It uses the appliances’ electrical signatures to infer their real-time usage (e.g., on/off states). However, it is inherently difficult to extract these signatures reliably from utility-side smart meters due to their large sampling intervals (e.g., once every five minutes or longer).

Many modern appliances and industrial equipments have highly non-linear electrical circuits that often lead to complex harmonics in the current waveforms they draw from the grid. Such non-linear loads are increasingly common in fact, from household appliances to industrial and office equipment. We observe that each combination of end loads that are present will create a unique pattern in the harmonics, which then serves as a fingerprint of the energy contained therein. This observation motivates us to use the fingerprint to estimate the states of specific end loads, thereby providing load disaggregation. The fingerprint has several advantages for power disaggregation compared with other kinds of electrical signatures. For example, it requires a relatively low sampling rate, hence is cheaper, than frequency electromagnetic interference (EMI) [3]. It can potentially provide a higher dimension of observability than active and reactive power signatures [4]. It can support flexible deployment scenarios by decoupling voltage and current channels [12]. Finally, it can be collected on a large scale for various deployment settings using self-powered sensor motes by harvesting energy from current transducer [2].

In this paper, we present an algorithm to detect the on/off states of electrical appliances reliably, using harmonic fingerprint signatures of current waveforms. Training the algorithm with comprehensive harmonic fingerprints is challenging, since the number of system states grows exponentially with the number of end loads present at the sub-branches. Training for all the possible state combinations is out of the question. The number of states that are important in practice could be much smaller, however. In a residential home, for example, the occupants’ daily usage patterns of their appliances, as well as the appliances’ inherent operational characteristics, will likely realize only a small subset of the possible states. This observation allows the harmonic signatures to be trained incrementally, using efficient semi-supervised learning with user input. Furthermore, the current waveforms obtained by the mote are often noisy, due to various real-world factors such as random phase shifts or disturbances introduced at analog-to-digital converters (ADCs). Hence, another challenge is to reliably estimate the states of end-loads based on their noisy harmonic signatures in an aggregated current waveform. A key contribution of this paper is the design of (i) a semi-supervised algorithm to learn a useful harmonic fingerprint map based on partial label information only, and (ii) an appliance state estimation algorithm that is robust against noise and errors in a natural deployment.

The balance of the paper is organized as follows. Section II presents the learning and detection algorithms for harmonic signatures. These algorithms are prototyped in a proof-of-concept testbed. Section III presents an empirical evaluation of the testbed, as well as simulation results driven by realistic data input. Section IV discusses related work. Section V concludes.
II. HARMONIC SIGNATURE LEARNING AND DETECTION

In this section, we present our algorithm for learning and detecting the harmonic signatures. For the learning, we develop a semi-supervised algorithm in which the harmonic fingerprints are incrementally trained using expectation maximization (EM) based on a Gaussian mixture model (GMM). In the algorithm, we first obtain a training data set for each individual appliance called a pivot data set. Then we generate a synthetic harmonic fingerprint map for each on/off state combination using the pivot data sets. Our EM algorithm starts with a synthetic harmonic fingerprint map as an initial guess. It then consistently improves its estimates as more training data or label information becomes available. Hence, users only need to supply pivot data sets to the system by switching the individual appliances on/off once in the beginning. Afterwards, the algorithm will continually learn the fingerprints with side information given occasionally by the users. For the appliance state detection algorithm we introduce a probabilistic model for quantifying the uncertainty of signatures, and use it to guide the search for the most probable sequences of state transitions in a hidden Markov model.

A. Parameter model for harmonic signature

Let us formulate a parameter model for the harmonic signatures. Let $f_0$ denote the operating grid frequency (e.g., $f_0 = 60\text{Hz}$ for U.S.). Assume that the aggregated current waveform is generated from a number $L$ of end-loads. We define the harmonic signature as a vector of spectral energies of the aggregated current waveform within bandwidth $W$, around each of the odd harmonics $\{2k - 1\}f_0 : k = 0, \ldots, M\}$ given the on/off state vector $X_L \in \{0, 1\}^L$. Let $Z_k$ denote the $k$th odd harmonic of the signature. We use a vector $Z = (Z_1, \cdots, Z_M)$ as a basis of the electrical signature given $X_L$.

We model the harmonic signatures sampled at a mote’s analog-digital-converter (ADC) as a probabilistic quantity, rather than a deterministic one, given the following observations. First, the spectral density of AWGN is constant in theory, but the spectral density becomes a random variable when it is sampled. It is because the AWGN bandwidth is infinite and the sampling theorem [11] suggests that the sampled noise cannot accurately represent the original noise. Second, the phase shift $\phi_i$ is not deterministic due to unstable voltage, current phase differences, and random phase shifts when the current waveform is sampled. Hence, we model $Z = (Z_1, \cdots, Z_M)$ as a random variable.

Consider a set of state vectors $\{\mathbf{x}_i \in X_L\}_{i=0, \ldots, S}$, where the state index $i$ is a decimal representation of the binary vector $\mathbf{x}_i$, and $S = 2^L - 1$. To simplify our model, we assume that $Z$ given $\mathbf{x}_i$ follows approximately a multivariate Gaussian distribution, as shown in (1).

$$P(Z|X_L = x_i) = \mathcal{N}(Z; u_i, \Sigma_i),$$

where $u_i$ and $\Sigma_i$ are an $1 \times M$ expectation vector and an $M \times M$ covariance matrix of $Z$, respectively, given $x_i$. Furthermore, we assume that the covariance matrix is a diagonal matrix, i.e., $(\Sigma_i)_{lm} = 0$ for $l \neq m$.

We can describe the parameters of the Normal distribution given $x_i$ by $M \times 1$ vectors of the expectations and variances as follows.

$$u_i = (\mu_{1|x_i}, \cdots, \mu_{M|x_i})^T, \quad d_i = (\sigma^2_{1|x_i}, \cdots, \sigma^2_{M|x_i})^T$$

where $\mu_{k|x_i}$ and $\sigma^2_{k|x_i}$ are the expectation and variance of $Z_k$ given $x_i$, and $d_i$ is a vector of the diagonal elements of $\Sigma_i$. The feature parameters of the harmonic fingerprint map are an $M \times S$ matrix of $U$ and $\Sigma$ in (2).

$$U = (u_0, \cdots, u_S), \quad D = (d_0, \cdots, d_S)$$

B. Greedy EM algorithm for harmonic signature training

Our system requires a training and pre-calibration process before running the disaggregation algorithm. During the training, we estimate the parameters $(U, D)$ from the following training data set: $\{(z_n, \pi_n)\}_{n=1, \ldots, N}$ where $\pi_n \in \{1, \cdots, S\}$ is the state index (or label) of the corresponding harmonic signature $z_n$. If we know the complete label information $\{\pi_n\}_{n=1, \ldots, N}$, the optimal parameter estimates for the Gaussian distribution can be readily found by

$$u_i = \frac{\sum_{n=1}^N \pi_n I_i(z_n)}{\sum_{n=1}^N \pi_n I_i(z_n)}, \quad d_i = \frac{\sum_{n=1}^N (z_n - u_i)^2 I_i(z_n)}{\sum_{n=1}^N I_i(z_n)}$$

where $I_i(y)$ is an indicator function such that $I_i(y) = 1$ if $x = y$, otherwise 0. However, it is not realistic to assume that users will collect a training data set with complete label information for all the possible states, since $S$ grows exponentially with the number of loads $L$. Instead, we assume that only a small fraction of the samples has label information. Let $DT_{Label}^L$ and $DT_{Label}^D$ denote the part of the data set with labels and the part without the labels, respectively, where $|DT_{Label}^L| \ll |DT_{Label}^D|$. Our training algorithm estimates the parameters $(U, D)$ from $DT_{Label}^L$ and $DT_{Label}^D$.

First, consider the training data without label information $DT_{Label}^D = \{z_n\}_{n=1, \ldots, N}$. The distribution of $z_n$ follows the Gaussian mixture model (GMM) in (4), where each sample $z_n$ is drawn from $x_i$ with probability of $\pi_i$ for $i = 1, \cdots, S$

$$p(z_n) = \sum_{i=0}^S \pi_i \mathcal{N}(z_n; u_i, \Sigma_i), \quad \sum_{i=0}^S \pi_i = 1$$

whose parameter is defined by $\Theta = (\Pi, U, D)$ and where $\Pi = (\pi_1, \cdots, \pi_S)$.

Meanwhile, some sample means and variances can be computed directly from $DT_{Label}^L$ by (3). Let $\bar{u}_k$ and $\bar{d}_k$ denote the sample mean and variance for $x_k$ given $DT_{Label}^L$. We assume that users perform a one-time training of each appliance by running it while the other appliances are off. Under this assumption, consider an $L \times M$ matrix whose ith row is the sample means of $\{z_n, 2^{i-1}\}$ (i.e., only appliance $i$ is on). We call the matrix a pivot mean matrix denoted by $U_p$. A pivot variance matrix can be similarly defined and is denoted by $D_p$.

Our overall greedy EM algorithm performs expectation maximization for the GMM multiple times in a greedy manner. To account for the information of $DT_{Label}^L$, we modify the standard EM algorithm slightly [1]. Figure 1 outlines this modified EM algorithm. It starts with an initial guess of the parameter $\Theta$. Then it runs two main computation steps (expectation and maximization, respectively) iteratively until
1) Initialization:
\[ U^0 = XU_P, \quad D^0 = XD_P, \quad \Pi^0 = S^{-1}1_{S \times 1} \]

2) Expectation Step:
\[ \gamma_{nk}^t = \mathcal{N}(z_n; u_k^t, \Sigma_k^t)\pi_k^t / \sum_{i=0}^{S} \mathcal{N}(z_n; u_i^t, \Sigma_i^t)\pi_i^t \]

3) Maximization Step:
\[ \Pi^{t+1} = \frac{1}{N^t} \sum_{n=1}^{N} \gamma_{nk}^t \quad \text{where} \quad N^t = \sum_{n=1}^{N} \gamma_{nk}^t \]
\[ u_k^{t+1} = \frac{1}{N_k^t} \sum_{n=1}^{N_k^t} z_n + \lambda_k \hat{u}_k \]
\[ d_k^{t+1} = \frac{1}{N_k^t} \sum_{n=1}^{N_k^t} (z_n - u_k^{t+1})^2 + \lambda_k \hat{d}_k \]

4) Evaluation:
\[ LLH^t = \sum_{n=1}^{N} \ln p(z_n|\Theta^{t+1}) \]

Fig. 1: Computation steps of expectation and maximization with partial label information

1) 1st EM Running: Set \( r = 1, N_{\text{NoImprv}} = 0 \)
Run EM Step 1,2,3,4 in Figure 1

2) Random Mean Parameter Search: \( \alpha \sim \mathcal{U}(\frac{1}{2}, 2), \beta = 0.3 \)
\[ U^0 \leftarrow \alpha U_r^*, \quad D^0 \leftarrow \beta U^0, \quad \Pi^0 = S^{-1}1_{S \times 1} \]

3) Greedy EM: Improve \( \Theta^{**} \)
\[ \Theta^{**} = \Theta^*_r \quad \text{if} \quad LLH^*_r > LLH^t \]
\[ N_{\text{NoImprv}} = N_{\text{NoImprv}} + 1 \quad \text{elseif} \]
Terminate if \( N_{\text{NoImprv}} > N_{\text{MaxNoImprv}} \)
Go back to 2 elseif.

Fig. 2: Greedy EM Algorithm

\[ \Theta^*_r = (\Pi^*_r, U^*_r, D^*_r) \]
\[ \Pi^* \]
\[ \pi_{ij} \]
\[ \gamma_{nk}^t \]
\[ \alpha \] the estimate of \( \Theta \) converges. Let \( \Theta^t = (\Pi^t, U^t, D^t) \) denote the GMM parameter at the \( t \)th iteration, where \( \Theta^0 \) denotes the initial guess of \( \Theta \). We use a synthesized harmonic signature for the initial guess such that \( U^0 = XU_P, \quad D^0 = XD_P \), where \( X \) is an \( S \times L \) matrix whose ith row vector \( X_i \) is \( x_i \).

During the expectation computation step, the algorithm computes the likelihood that \( z_n \) comes from \( x_k \) given \( \Theta = \Theta^t \), which is \( \gamma_{nk}^t = P_F(z_n|x_k, \Theta^t) \). During the maximization step, the algorithm updates parameter \( \Theta^{t+1} \) for the next iteration.

The update is done by re-estimating \( \Theta^{t+1} \) using samples \( \{z_n\} \) weighted by likelihoods \( \gamma_{nk}^t \). To account for the side information from \( DT_{\text{Label}} \), the algorithm replaces \( (u_k^{t+1}, d_k^{t+1}) \) with \( \left( \hat{u}_k, \hat{d}_k \right) \), if the number of labeled samples is greater than \( N_{\text{min}} \) for \( x_k \), i.e., \( \lambda_k = 1 \) if \( \sum_{n=1}^{N} I_k(l_n) \geq N_{\text{min}} \) for \( z_n \in DT_{\text{Label}} \), otherwise \( \lambda_k = 0 \). The EM algorithm iterates these two steps until \( LLH^t \) (the log likelihood of \( z_{1:n} \)) converges in the evaluation step.

Although the EM algorithm is guaranteed to converge, its estimate is likely to be a local maximum. To alleviate the problem, our overall algorithm runs the expectation maximization in Figure 1 multiple times in a greedy manner. Figure 2 specifies this greedy EM algorithm. Let \( LLH^*_r \) and \( \Theta^*_r \) denote the final estimate of the parameter. First, the greedy EM algorithm finds the local maximum of \( \Theta \) by running the expectation maximization in Figure 1 (in 1). Then it explores the neighborhood parameters randomly, by running the expectation maximization again, setting \( U^0 \) to \( \alpha U_r^* \) (in 2 and 3). The random variable \( \alpha \) is drawn from a uniform distribution of \((\frac{1}{2}, 2)\). It updates \( \Theta^* \) only if \( LLH^*_r \) is improved (in 4). The procedure is repeated until we reach the runtime budget of the number of iterations.

C. State estimation using hidden Markov model

We design an algorithm for sequentially detecting the on/off states of appliances using a hidden Markov model (HMM) and forward dynamic programming (DP). Let us define the following probability functions: \( \pi_i = P\{X_L = x_i\} \), \( r_{ij} = P\{X_L = x_j|X_L = x_i\} \), and \( \theta_i(z) = P\{Z = z|X_L = x_i\} \), where \( i = 0, \ldots, S \) is a decimal representation of the binary vector \( x_i \). We assume that the state transitions of the individual end loads follow a Markovian process given by \( p(X_L^t|x_L^0, \ldots, x_L^{t-1}) = p(x_L^t|X_L^{t-1}) \).

Now we can describe the vector of all the parameters \( \Theta = \{\pi, \pi_i(z), r_{ij}\}_{i,j} \) by the HMM illustrated in Figure 3. Consider the probabilities of individual loads changing their states between the mote’s two consecutive wake-up instants, such that \( q = P\{x_i = 1 \rightarrow 0 \} \). The transition probability is \( r_{ij} = q^d_{ij}(x_i, x_j) \), where \( d_{ij}(x, y) \) is a Hamming distance between \( x \) and \( y \). Note that the specification of the transition probability is particularly important in our HMM formulation for the sake of robustness. We specify the transition probability \( r_{ij} \) by finding an appropriate model for \( q \). Let us define \( \delta_x = ||z^{(t)}|| - ||z^{(t-1)}|| \). Intuitively, \( q \) will approach 0 (or 1) when \( \delta_x \) approaches 0 (or \( \infty \)). Let \( \delta_{\text{min}} \) and \( \delta_{\text{max}} \) denote the minimum and maximum values of \( \delta_x \), respectively, and \( \epsilon > 0 \) represent a near-zero probability. Then we consider the following four different types of \( q \):

- **(a) NoQ:** \( q = 1 \);
- **(b) ConstQ:** \( q = 0.5 \);
- **(c) ThresholdQ:** \( q = \epsilon \) if \( \delta_x < \delta_{\text{min}} \), \( q = 0.5 \) otherwise; and
- **(d) LogisticQ:** \( q(\delta_x) = \frac{1}{1 + \exp((a+b)\delta_x)} \).

For NoQ, \( r_{ij} = 1 \) for all the possible state transitions, so the HMM is effectively not used. For ConstQ, we assume that all the individual end loads have an equal probability of changing their states regardless of \( \delta_x \). In ThresholdQ and LogisticQ, we consider that \( \delta_x \) is proportional to the real power change by Parseval’s theorem. Especially, LogisticQ uses a logistic distribution to model a smoothly changing probability between 0 and 1 given \( \delta_x \). The coefficients \( (a, b) \) are estimated from two pairs of samples, \((\delta_x^\text{min}, \epsilon)\) and \((\delta_x^\text{max}, 1-\epsilon)\). We use LogisticQ for \( q \) in our HMM formulation.
of binary states (TV, LT, HD, MN, FG) where the first and last bits are the most and least significant bits, respectively. We collect 5000 samples of z that are randomly chosen from the 32 states in the testbed. Figure 4 shows samples of the first three out of 10 harmonics with sample means of the 32 states in a 3-D scatter plot. The plot shows that the harmonic samples roughly follow a multivariate Gaussian distribution with the diagonal covariance matrix in (1).

We use Kullback-Leibler (K-L) divergence as a main performance metric for the greedy EM algorithm. The metric is commonly used in statistics as a measure of the similarity between two density distributions. Let Θ and Θ̂ denote the GMM parameter and its estimate by the greedy EM algorithm, respectively. The KL divergence of GMM with Θ̂ from Θ is defined by

$$KL(p_Θ(z)||p_{Θ̂}(z)) = \int_z p_Θ(z) \ln \frac{p_Θ(z)}{p_{Θ̂}(z)} dz$$

where $p_Θ(z)$ is the GMM distribution given parameter Θ. Note that no closed form expressions exist for the above formula. Instead, the KL divergence is computed indirectly by Monte Carlo sampling [5]. A smaller $KL ≥ 0$ implies a more accurate Θ̂, hence a better estimate for the parameter of harmonic fingerprint (U, D).

We run the greedy EM algorithm on 5000 samples with $N_{sample} = 30$. Figure 5 compares the algorithm’s performance with that of standard EM, greedy EM without label information, and greedy EM with 10% label information (i.e., the labels of 500 samples are known). The four plots in the figure show the actual sample means (rectangular marks) and their estimates (star marks) for the first 3 harmonics, with the KL divergence given on top of the plot. The leftmost plot shows that the initial guess, i.e., a synthetic harmonic map, estimates the GMM distribution poorly, resulting in $KL = \infty$. The second plot corresponds to the standard EM algorithm; the K-L divergence is $KL = 0.92$. The third plot shows that our greedy EM algorithm can improve the training performance significantly, by reducing the K-L divergence to $KL = 0.48$ from the $KL = 0.92$ in the second plot. Finally, the rightmost plot shows that the greedy EM algorithm estimates the GMM parameter almost exactly, with only 10% of label information available, resulting in $KL = 0.07$.

### B. Simulation Results

We now evaluate the state estimation performance of our algorithm for different noise levels through simulations. To drive the simulations, we generate a synthetic trace of the appliance states over five months, using a multidimensional semi-Markov chain in which each appliance stays for four hours in the off state and one hour in the on state on average. In each state i, at simulated time t, we randomly choose 800 continuous samples of $cw_i[t]$ (corresponding to approximately 10 cycles of the waveform) within one minute of an actually measured aggregated current waveform. We add zero-mean white Gaussian noise $n_σ \sim N(0, σ)$ to $cw_i[t]$ and increase noise level σ to evaluate the robustness of our state estimation algorithm over Noise-to-Signal Ratio (NSR) defined by $\frac{σ}{\text{std}(cw_i[t])}$. Note that we use the metric NSR instead of a conventional metric Signal-to-Noise ratio (SNR) because the signal power is an uncontrollable parameter in the context of our application.
We record the state estimation performance while varying the NSR from 0 to 1 for different settings of the HMM, i.e., NoQ, ConstQ, ThresholdQ, and LogisticQ.

Figure 6(a) shows the ground truth states over a day and the corresponding estimates, for the different HMMs at $NSR = 0.8$. In the figure, NoQ shows a large estimation error of 8.5% at $NSR = 0.8$. The estimation error is drastically reduced to 1.11% and 0.13% with ThresholdQ and LogisticQ, respectively, relative to NoQ. Note that, on the other hand, ConstQ achieves little performance gain (by about 1%) compared to NoQ. It is because employing $\delta_k$ in the HMM helps to successfully suppress unlikely state transitions, through higher transition costs according to (6). Figure 6(b) shows the BER performance as the NSR varies from 0 to 1. It shows that our HMM under LogisticQ is highly robust to low NSRs, providing over 10 times better performance compared with NoQ.

C. Experimental Results

We now evaluate our testbed with five appliances. In the experiments, the first five harmonics are chosen for the multivariate Gaussian signature model in (1). In each experiment, the on/off states of the appliances are automatically controlled by the SPlug meter except for the fridge, which has its own duty cycles driven by a built-in controller. The on/off transitions are independently generated for the individual appliances under SPlug control every 20 seconds on average according to the Exponential distribution. Figure 7 evaluates the accuracy of the state estimation and the per-load energy breakdown. We use the bit error rate (BER) as the metric for the state estimation. It is defined for the $l$th load by

$$\text{BER}_{l} = \frac{1}{v} \sum_{t=1}^{v} \left| x_{t}^{(l)} - x_{t}^{*\text{est}(l)} \right|$$

where $x_{t}^{(l)}$ and $x_{t}^{*\text{est}(l)}$ is a groundtruth binary state of $l$th load at $t$th time index of the observation time $v$ and its estimate. As such, the BER gives the ratio of erroneously estimated on/off states to the total number of states observed for the load.

We use the state estimation algorithm with NoQ (i.e., without HMM) as a baseline for showing the robustness of the proposed HMM approach, i.e., the algorithm with LogisticQ. First, we conduct an one-hour of measurements for the five appliances. The aggregated current waveform is sampled for 10 cycles (i.e., approximately 100ms) every two seconds by the mote. Figure 7 shows our measurement results. Note that some appliances often enter an idle mode or make a
state transition that generates unexpected harmonic signatures during the experiment. For example, the laptop can enter various idle levels during operation. Furthermore, the fridge creates large spikes in the harmonics after it is turned on and until its compressor stabilizes. As a result, a large number of consecutive errors in state estimation are observed for NoQ whereas the HMM under LogisticQ estimates individual loads’ consumption robustly.

For the five appliances, as shown in Figure 7, our algorithm achieves a BER of 4% in the state estimation.

IV. RELATED WORK

There has been significant work on indirect metering approaches that employ sophisticated signal processing algorithms to infer the energy usage of individual end loads from aggregate consumption measurements. Hart et al. developed a non-intrusive load monitoring (NILM) system [4] that monitors the total electrical load only, then disaggregates it by detecting the time-domain signatures of individual appliances as step changes in the aggregated consumption. Marchiori et al. [10] proposed a circuit-level load monitoring that uses a two-dimensional probability map of the total active and reactive power consumption for all possible combinations of the individual appliances that are active. A similar approach was proposed by Kim et al. that uses a hidden Markov model (HMM) and an expectation maximization (EM) algorithm for aggregated real power measurements in homes [8].

Signal processing techniques in the frequency domain have also been utilized to identify the power states of specific appliances. The transient event detector has been proposed to identify the on/off events of a specific type of appliances, by analyzing relevant transient events of the electrical loads [9]. Patel et al. proposed an algorithm to infer the on/off states of a particular appliance, by detecting its unique signature of electromagnetic interference (EMI) in power line voltage in the frequency domain [3]. Using harmonic signatures for device identification has been studied [13]. The authors use neural network classification models to classify devices in operation for the sake of power-quality monitoring. They use the Fluke 41 power harmonics analyzer, which is an expensive high-end measurement device, to collect high-quality harmonic data, and test their classification algorithm for 10 devices in an off-line manner. Rather than device classification, we employ harmonic signatures for inferring the energy consumption of the devices or appliances. Furthermore, to reduce the deployment cost, we opt for a sensor network of inexpensive power meters and mote sensors instead of sophisticated harmonics analyzers. As a result of this decision, we develop estimation techniques that are resilient against the more noisy and unreliable harmonic signatures in our deployment context. We report testbed experiments to illustrate the performance of our disaggregation system. We have demonstrated a compact setup of our experimental testbed [6].

V. CONCLUSION

We have proposed a new approach of learning and detecting individual appliance usage utilizing aggregated current waveforms sampled by mote sensors. Our experiments, using simulations and a real testbed, show that the proposed system operates reliably in the face of noise in real deployment environments. Our key contributions are the development of an algorithm to learn the harmonic fingerprint map efficiently, and the development of a hidden Markov model to reliably estimate appliance states under unreliable harmonic signatures.

The proposed research is potentially applicable to the health monitoring of HVAC systems and industrial equipment such as pumps, chillers, and fans, which produce distortions in current waveforms under mechanical faults. It is interesting for future research to make use of harmonic signatures to inform repair and maintenance decisions in buildings including factory environments.

ACKNOWLEDGMENTS

This research was supported in part by the Singapore Agency for Science, Technology, and Research (A*Star), under a grant award for the Human-Centered Cyber-Physical Systems Programme at the Singapore Advanced Digital Sciences Centre, and in part by the National Natural Science Foundation of China (NSFC) under grant number 61429301.

REFERENCES


